Mining Optimal Actions for Profitable CRM

Charles X. Ling Tielin Chen
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
E-mail: {ling, tchen}@csd.uwo.ca

Qiang Yang
Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon Hong Kong
E-mail: qyang@cs.ust.hk

Jie Cheng
Global Analytics
Canadian Imperial Bank of Commerce (CIBC)
Toronto, Ontario, Canada M5J 2S8
E-mail: jie.cheng@CIBC.ca

Abstract

Data mining has been applied to CRM (Customer Relationship Management) in many industries with a limited success. Most data mining tools can only discover customer models or profiles (such as customers who are likely attritors and customers who are loyal), but not actions *that would improve customer relationship (such as changing attritors to loyal customers). We describe a novel algorithm that suggests actions to change customers from an undesired status (such as attritors) to a desired one (such as loyal). Our algorithm takes into account the cost of actions, and further, it attempts to maximize the expected net profit. To our best knowledge, no data mining algorithms or tools today can accomplish this important task in CRM. The algorithm is implemented, with many advanced features, in a specialized and highly effective data mining software called Proactive Solution.*

1 Introduction

There are two aspects for Enterprises to build a strong CRM (Customer Relationship Management). One is "enabling CRM", which focusses on the infrastructure, database management, multiple touch-point information integration, and system integration. That is, enabling CRM facilitates and enables the basic functionality of CRM. The other aspect is "intelligent CRM", which emphasizes on making better decisions on improving customer relationship based on customer data. Data mining has been applied to intelligent CRM with a limited success.

A common problem in current applications of data mining in intelligent CRM is that people tend to focus on, and be satisfied with, building up the models and interpreting them, but not to use them to get profit explicitly. More specifically, most data mining algorithms (predictive or supervised learning algorithms) only aim at constructing customer profiles, which predict the characteristics of customers of certain classes. For example, what kind of customers (described by their attributes such as age, income, etc.) are likely attritors (who will go to competitors), and what kind are loyal customers? This knowledge is useful but it does not directly benefit the Enterprise. To improve customer relationship, the Enterprise must know what *actions* to take to change customers from an undesired status (such as attritors) to a desired one (such as loyal customers). To our best knowledge, no data mining algorithms or tools have been published or are available to accomplish this important task in intelligent CRM.

The task is not easy. First of all, actions cost money to the Enterprise. A customer of an insurance company could be given a new car (action) in exchange of the policy renewal (from possible attritor to loyal customer), but it is clearly not worthwhile. Therefore, one must take into account the cost of actions to the Enterprise. Second, customers are different in their values to the Enterprise. An action worthwhile to one customer may not be worthwhile to another. Third, many actions are possible but which ones are optimal? The key question is what actions are best to each different customer such that the potential benefit of taking these actions is optimal (after taking into account the cost of actions).

In this paper, we will describe a novel procedure that utilizes decision-tree models to find optimal actions to take to change customers from the undesired status to the desired one while maximizing the expected net profit (after taking away the cost of actions). We will also describe our data mining software called Proactive Solution that implements the algorithm, along with many advanced features. Applications of Proactive Solution will be briefly discussed.

2 Building Decision Trees for Actions

We describe a new data mining system that utilizes decision tree to discover actionable solutions for the status change problem in CRM. The algorithm is implemented in a data mining system called "Proactive Solution", a data mining software for intelligent CRM.

The overall process of Proactive Solution can be briefly described in the following four steps:

1. Import customer data: data collection, data cleaning, data pre-processing, and so on.

2. Build customer profiles: using an improved decision-tree learning algorithm [7] to build customer profile from the training data.

3. Search for optimal actions for each incoming customer (see Section 2.1 for details). This is the key and novel component of our data mining system Proactive Solution.

4. Produce reports for domain experts to review the solutions and selectively deploy the actions.

In the next subsection, we will mainly discuss components of the step 3 (search for optimal actions) in details.

2.1 Search for Optimal Actions

The basic idea for searching optimal actions in decision tree is quite simple. After a customer profile is built, the resulting decision tree can be used to classify, and more importantly, give probability of customers in the desired status (such as being loyal or high-spending). When a customer (can be either an training example used to build the decision tree or an unseen testing example) falls into a particular leaf with a certain probability of being in the desired status, the algorithm tries to "move" the customer into other leaves with higher probabilities of being in the desired status. The probability gain can be converted into an expected gross profit. However, moving a customer from one leaf to another means some attribute values of the customer must be changed. The attribute value changes are viewed as actions, and actions incur costs. The algorithm searches all leaves in the tree to find a best leaf to move the customer to such that the gross profit minus the cost of the corresponding actions is maximal.

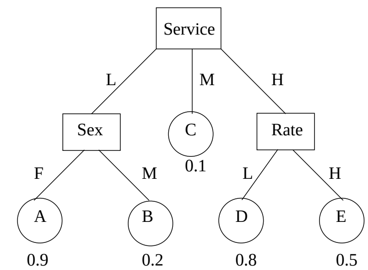

Figure 1. An example of customer profile

Here is an overly simplified example to show the working of the algorithm. Figure 1 represents a hypothetical decision tree as the customer profile of loyal customers built from a bank. The tree has five leaf nodes (A, B, C, D, and E), each with a probability of customers' being loyal. The probability of attritors is simply 1 minus this probability.

Let say a customer, Jack, with Service (service level) being L (low), Sex being M (male), and Rate (mortgage rate) being L, is classified by the decision tree. Clearly, Jack falls into the leaf B, which predicts that Jack will have only 20% chance of being loyal (or Jack will have 80% chance to churn in the future). The algorithm will now search through all other leaves (A, C, D, E) in the decision tree to see if Jack can be "replaced" into a best leaf with the highest net profit.

1. Consider leaf A. It does have a higher probability of being loyal (90%), but the cost of action would be very high (Jack should be changed to female), so the net profit is a negative infinity.

2. Consider leaf C. It has a lower probability of being loyal, so the net profit must be negative, and we can safely skip.

3. Consider leaf D. There is a probability gain of 60% ($80\% - 20\%$) if Jack falls into D. The action needed is to change Service from L (low) to H (high). Assume that the cost of such a change is \$200 (given by the bank). If the bank can make a total profit of \$1000 from Jack when he is 100% loyal, then this probability gain (60%) is converted into \$600 (1000×0.6) of the expected gross profit. Therefore, the net profit would be \$400 ($600 - 200$).

4. Consider leaf E. The probability gain is 30% ($50\% - 20\%$), which transfers to \$300 of the expected gross profit. Assume that the cost of the actions (change Service from L to H and change Rate from L to H) is \$250, then the net profit of moving Jack from B to E is \$50 ($300 - 250$).

Clearly, the node with the maximal net profit for Jack is D, with suggested action of changing Service from L to H.

Notice that actions suggested for customer status change imply only correlations (not causality) between customer features and status. Like other data mining systems, the results discovered (actions here) should be reviewed by domain experts before deployment. This is the Step 4 discussed at the begining of this Section.

The algorithm for searching the best actions can thus be described as follows: for each customer, search every leaf node in the decision tree to find the one with the maximum net profit using the formula:

$$P_N = P_E \times P_{gain} - \sum COST$$

where P_N denotes the net profit, P_E denotes the total profit of the customer in the desired status, P_{gain} denotes the probability gain, and $COST$ denotes the cost of each action involved.

In the following subsections, several features of Proactive Solution are described in more details.

2.2 Cost matrix

Attribute value changes will incur costs in most cases, and such costs can only be determined by domain knowledge and/or domain experts. For each attribute used in the decision tree, a cost matrix is used to represent such costs. Users of Proactive Solution must provide values in the cost matrix. In most domains, values of many attributes (such as sex, address, number of children,etc.) cannot be changed with any reasonable amount of money. Those attributes are called "hard attributes". In this case, users must assign a very large number to every entry in the cost matrix. This would naturally prevent Proactive Solution from suggesting any changes on the hard attributes. If some value changes are possible with reasonable costs, then those attributes (such as the Service level, Rate, promotion packages, etc) are called "soft attributes". Note that the cost matrix needs not to be symmetric. One can assign \$200 as the cost of changing service level from low to high, but infinity (a very large number) as the cost from high to low, if the bank does not want to "degrade" service levels of customers as an action.

One might ask why hard attributes should be included in the tree building process in the first place, since they can prevent customers from being moved to other leaves. This is because that many hard attributes are important in accurate probability estimation of the leaves. When the probability estimation is inaccurate, the reliability of the prediction would be low, or the error margin of the prediction (see Section 2.4) would be high. In addition, even if a customer falls into a leaf with some hard attributes on the path from the root to the leaf, the customer can still be moved to other leaves where the hard attributes have the same values, or the hard attributes are irrelevant. The example given in Figure 1 is such a case. Customer Jack falling into leaf B can be moved to leaves D or E without changing the hard attribute "sex".

One might argue that the cost of attribute value changes is hard to give. Exactly how much does it cost to a bank to open a new loan account? To address this problem, we allow users to input action costs in a fuzzy term in the format of *(mean, deviation)*: users can specify the mean and the deviation of the mean of the cost. Proactive Solution will calculate lower and upper bounds of the cost according to the mean, deviation, and the confidence level given by the users (see Section 2.4). Note also that all costs are relative; exact amounts are not important for obtaining optimal actions of each customer.

2.3 Building Multiple Decision Trees

Another improvement we have made in Proactive Solution is to build multiple trees using the same training data but with different subsets of hard attributes (all soft attributes are included). Figure 2 shows two decision trees with different hard attributes. As discussed in Section 2.2, hard attributes do sometimes prevent customers from being moved to other leaf nodes. Trees with different hard attributes provide more chances for customers to be moved to leaves with positive net profits. For each customer, the optimal actions are taken from the best tree with the highest net profit. Experiments show that Proactive Solution with multiple trees often doubles the total sum of net profits of all customers compared to a single decision tree.

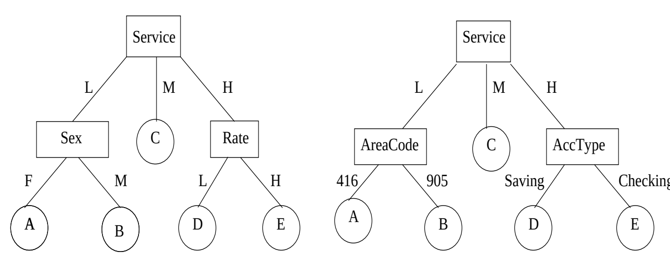

Figure 2. Multiple trees with different subsets of hard attributes.

2.4 Error Margin

To produce realistic solutions, we used a sophisticated statistical method to calculate the reliability of the solution, measured by error margins. The error margin is related to the confidence level (set by the users) of the results, the accurate probability estimation of the decision tree, and the number of examples falling into the leaves. For example, if the confidence level is set to 95%, and Proactive Solution predicts a net profit of $800 with an error margin of $200 for a particular customer, then with probability 95%, the actual net profit would be within $600 ($800 - 200$) and $1000 ($800 + 200$).

Section 2.2 discussed reasons for including hard attributes for reducing the error margin. If all hard attributes are excluded in the decision tree, the error margin can be very large, and the lower bound of the net profit can be small or even negative, making the benefit of taking the actions uncertain.

3 Applications of Proactive Solution

We have implemented the novel action-searching algorithm and features discussed in the previous subsections in Proactive Solution. Many other advanced features have been implemented but are not discussed here due to space limitation.

Proactive Solution has been applied to various intelligent CRM tasks in financial institutions and insurance companies with satisfactory results. One task is to promote the purchasing of financial products (from low-spending to high-spending). The dataset contains about 100 attributes. The hard attributes include customer personal and demographic information. The soft attributes include account types, fee charges, agent information (such as agents experience, agent management style, etc.), other products, promotional information, etc. Proactive Solution increases substantially the total spending of customers when compared to a control group of customers.

Proactive Solution is a software for mass customization in CRM, since actions for different customers can be different. It is action-oriented since it suggests actions needed for improving CRM. It is proactive, since it suggests actions before the situation is getting worse. For example, Proactive Solution suggests actions to prevent customers from leaving before they actually leave. It is profit-driven since it aims at maximizing the net profit for the Enterprise (instead of some data mining evaluation measure such as error rate or lift). It is highly effective since it deploys many advanced features to accomplish this task extremely well.

4 Summary

Intelligent CRM improves customer relationship from the data about customers. Unfortunately, very little work has been done in data mining on how to improve (actions) such relationship of customers (changing customers from an undesired status to a desired one). Proactive Solution is the first such system that proposes proactive actions while maximizing the net profit. It offers effective solutions to intelligent CRM of any Enterprises.

References

[1] Stephen J. Smith Alex Berson, Kurt Thearling. *Building Data Mining Applications for CRM*. McGraw-Hill, 1999.

[2] Jill Dyche. *The CRM Handbook: A Business Guide to Customer Relationship Management*. Addison-Wesley, 2001.

[3] Barton J. Goldenberg. *CRM Automation*. Prentice Hall, 2002.

[4] C. X. Ling and H. Zhang. Toward Bayesian classifiers with accurate probabilities. In *Proceedings of The Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Springer, 2002.

[5] Tom Mitchell. Machine learning and data mining. *Communication of the ACM*, 42(11), November 1999.

[6] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453. Morgan Kaufmann, 1998.

[7] J.Ross Quinlan. *C4.5 Programs for machine learning*. Morgan Kaufmann, 1993.

[8] Padhraic Smyth Usama Fayyad, Gregory Piatetsky-Shapiro. From data mining to knowledge discovery in databases. *AI Magazine*, 17(11), Fall 1996.